Journal of Computer Science and Artificial Intelligence
ISSN: 3078-8242 (Print), ISSN: 3078-2309 (Online) | Vol. 4, No. 3, 2025

An Empirical Study on Used Car Price Prediction Using Supervised and Unsupervised Learning

Yutao Rao [1], Yang Li [2], Haotian Wu [3]

[1] University of Illinois Urbana-Champaign, Champaign, United States

[2] Xi'an Jiaotong-Liverpool University, Suzhou, China

[3] Imperial College London, London, United Kingdom

Abstract: This project focuses on predicting used car prices using a combination of data preprocessing, unsupervised learning, and advanced supervised learning techniques [1, 2]. The goal was to develop an accurate model for price prediction by exploring patterns in vehicle features and leveraging robust machine learning methodologies. The dataset was meticulously cleaned and enhanced by imputing missing values using a random forest-based imputation method [3], splitting multi-dimensional features such as engine specifications, and categorizing variables like brand and transmission types. Principal Component Analysis (PCA) was employed to reduce dimensionality, retaining 95% of the dataset's variance, and unsupervised clustering algorithms, including K-Means, K-Modes, and hierarchical clustering, identified meaningful groupings that provided insights into vehicle segmentation. For supervised learning, we implemented and compared multiple models, including Elastic Net regression, Random Forest, Support Vector Machines, and XGBoost. The XGBoost model demonstrated superior performance with an R^2 of 0.87 and a MAPE of 21.07%, effectively capturing non-linear relationships [1, 2]. Key predictors, including mileage, horsepower, model year, and brand grouping, provided important prediction power into price drivers. In the open-ended question part, we estimated the original prices of cars as if they were brand new using Random Forest and XGBoost models, adjusting attributes such as mileage and model year to simulate new conditions. Since our predicted prices closely fit with official release prices, so we successfully used machine learning techniques to achieve accurate predictions for car prices.

Keywords: Used Car Price Prediction, Machine Learning, XGBoost, Random Forest, Data Preprocessing, Feature Engineering.

1. Literature Review

1.1. Car Price Prediction Using Machine Learning Techniques [2]

https://www.ceeol.com/search/article-detail?id=746689

1.1.1. Hybrid Approach Description

The most interesting approach this article used is the hybrid method that integrates Random Forest (RF), Support Vector Machines (SVM), and Artificial Neural Networks (ANN) to enhance the accuracy of car price prediction.

Random Forest (RF): Used for its ability to handle high-dimensional data and capture complex feature interactions. Helped mitigate overfitting by averaging predictions across multiple decision trees.

Support Vector Machines (SVM): Effective for classification tasks, particularly in distinguishing between price categories (e.g., cheap vs. expensive cars). Leveraged kernel functions to handle non-linear relationships between features and target variables.

Artificial Neural Networks (ANN): Modeled non-linear and complex dependencies in the dataset. The ANN component provided flexibility in learning hierarchical patterns within the data.

1.1.2. Stacking for Ensemble Learning

The three models were combined using a stacking technique:

Individual predictions from RF, SVM, and ANN were treated as inputs to a meta-model.

The meta-model produced the final prediction, enhancing overall accuracy by leveraging the strengths of each base model.

1.2. Used Car Price Prediction Using Machine Learning: A Case Study [1]

https://ieeexplore.ieee.org/abstract/document/9800719

1.2.1. Handling Skewness in Data

Issue Identified: The target variable, price, exhibited a right-skewed distribution, with a concentration of lower-priced cars and a few high-priced outliers.

Solution: A log transformation was applied to the price variable, which effectively:

Reduced the skewness, resulting in a more symmetric distribution.

Improved the linear relationship between the target variable and predictors, enhancing the performance of regression models.

Impact: Log transformation significantly stabilized variance, ensuring better generalization during model training and testing.

1.2.2. Gradient Boosting Regressor (GBR)

Model Overview:

GBR is an ensemble learning method that builds sequential decision trees, where each tree corrects the errors of the previous ones.

It uses a learning rate to control the contribution of each tree, preventing overfitting.

Why GBR Stood Out:

It effectively captured complex, non-linear relationships in the dataset.

Its ability to weigh features dynamically during training led to better handling of interactions among predictors.

2. Data Processing and Summary Statistics

2.1. Extracting Engine Information

The engine column initially included one field covering information including engine size, cylinder count, and horsepower. This data was split into three separate attributes using the stringer package that could be better examined: patterns starting with "V" and numbers were used to get cylinder count; patterns ending in "L" or "Liter" were used to find engine displacement. This modification made it feasible to include more exact and practical tools, therefore enhancing the dataset quality for next studies. After splitting, there are three variables, all numerical. Engine_liter displays the engine's size in liters; horsepower indicates the engine's power capacity; cylinders show the engine's cylinders count. Predictive models depend much on these numerical factors since they provide vital information on vehicle performance and efficiency.

2.2. Imputing Missing Details in Engine and Accident

Key variables—cylinders, engine_liters, horsepower, accident histories—had missing values in the sample; missing rates ranged from 5% to 20%. We used MissForest, a strong random forest-based imputation technique applied with the MissForest package in R, to handle this problem. Using random forest models, MissForest repeatedly forecasts missing values by capturing intricate interdependencies among attributes and managing heterogeneous data types (Stekhoven & Bühlmann, 2012) [3]. MissForest shines in maintaining contextual linkages and guaranteeing realistic replacements when compared to more basic methods such as mean or regression-based imputation. For missing accident records, for instance, characteristics including model year, price, and fuel type were imputed; engine liters and cylinders helped project reasonable horsepower figures.

Out-of- bag (OOB) error estimates were used to validate the accuracy of this imputation technique and find that the filled values fit expected patterns rather nicely. For example, assumed horsepower figures regularly showed logical connections with engine_liters and cylinders. This method guaranteed that for next modeling projects the dataset stayed analytically strong and dependable.

2.3. Cleaning and Standardizing Price and Mileage

Originally, the price column showed monetary values—such as $25,000 or $3,000—formed as strings with dollar signs ($) and commas (,). These components make modeling or numerical computations impossible from the column. We eliminated the non-numeric characters—dollar signs and commas—then translated the cleaned text into numerical form facilitating direct numerical operations for preparation for analysis. Likewise, the mileage column included numbers mixed with commas and text descriptors like "mi." for miles, with values like "12,345 mi." These text components had to be eliminated to bring the data into line. I eliminated the text labels and commas and, therefore left just the numerical value of every entry. Following cleaning, the column was converted into a numerical form that was fit for use in modeling and consistent comparison of vehicle utilization. Using these changes, we transformed messy, text-based forms of pricing and milage into neat, numerical ones easily examined.

2.4. Refining Fuel Type

Online research helped to find the appropriate fuel type for particular car models, therefore addressing missing values in fuel types. We built a mapping table from trustworthy sources and filled in the absent entries depending on the matching brand and model. This method guaranteed correctness and consistency by using outside data to preserve dataset dependability free from assumption introduction.

2.5. Addressing Clean Titles

Missing entries abound in the clean_title field, a categorical feature with "Yes" and "No" as potential values. Reviewing the data, we found that every non-missing value had a "Yes" label. This discovery led us to assume that the missing entries most certainly matched "No." We replaced all blank or missing items with "No," therefore guaranteeing consistency and completeness of the dataset. This method was based on the presumption that, whilst its absence suggests otherwise, a clean title is expressly indicated when present. This approach helped us to preserve the integrity of the categorical variable by managing the missing values, therefore avoiding unjustified presumptions or prejudice.

2.6. Removing Redundant and Incomplete Data

To ensure data quality, rows with partial values in significant columns including transmission, ext_col, and int_col were removed. Given low missing values in these columns, exclusion was a fair choice that preserved the general integrity of the dataset without considerably reducing the sample size. Moreover, the engine column was thrown away after its key data split into many variables.

2.7. Categorizing Transmission Types

We divided the transmission variable into several groups including "Automatic," "Manual," "CVT," "Single-Speed," "Dual-Clutch," and "Multi-Speed," so organizing it more structurally and analyzable. Applying a bespoke function that found keyword patterns in the raw transmission data helped one to achieve this classification. For example, "Automatic" or "A/T" was mapped to "Automatic," whereas "Manual" or "M/T" was assigned "Manual." Any kind of transmission that fell outside of these stated categories was classified as "Other." By streamlining the variable, this standardization made it simpler to investigate correlations between other dataset characteristics and transmission kinds.

2.8. One-Hot Encoding for Low-Cardinality Categorical Variables

To prepare for supervised learning, we applied one-hot encoding to categorical variables with fewer levels, such as fuel_type, accident, clean_title, and transmission. This step converted these variables into dummy variables, enabling them to be effectively used in modeling while preserving their categorical information.

2.9. Removing Outliers

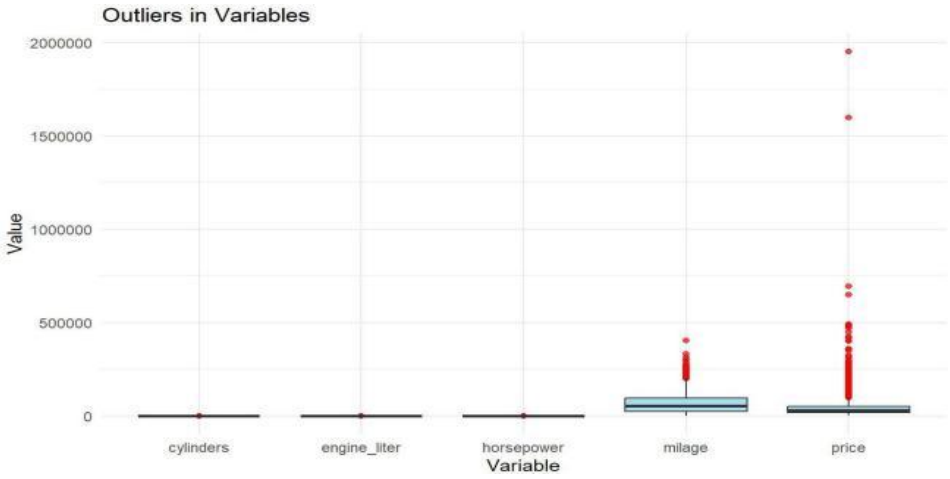

Figure 1. Boxplot of Outliers in Key Numerical Variables

We observed a few extreme values across important numerical variables including cylinders, engine_liter, horsepower, milage, and price throughout the data cleaning procedure. We deleted outliers depending on the boxplot analysis to handle this. We specifically found values for each variable that deviated greatly from the usual range using the interquartile range (IQR). We sought to guarantee that the data stayed typical of the larger dataset by removing these extreme points, therefore reducing the influence of any abnormalities. This change produced a more dependable and tidy dataset for study.

2.10.Summary Statistics

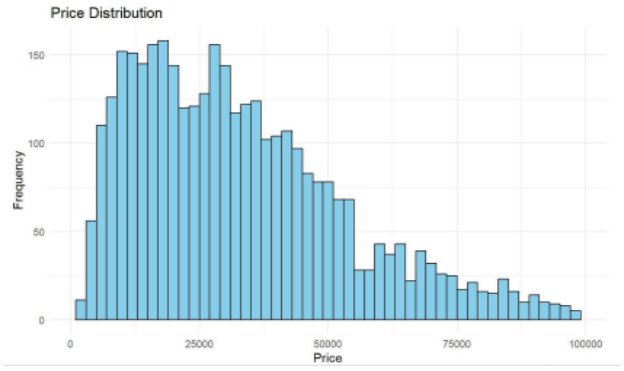

Figure 2. Price Distribution

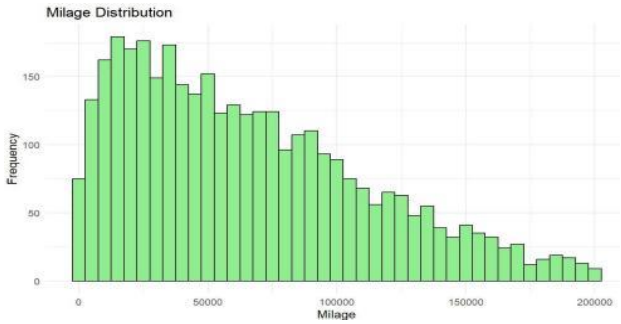

Figure 3. Mileage Distribution

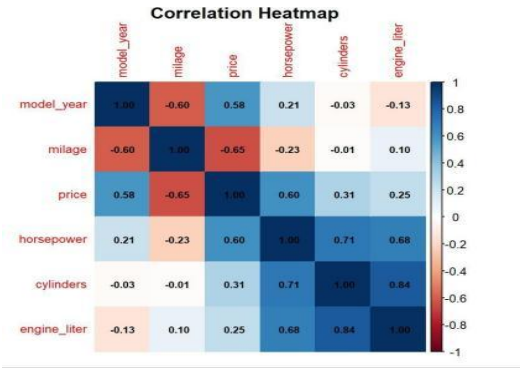

Figure 4. Correlation Heatmap of Numerical Variables

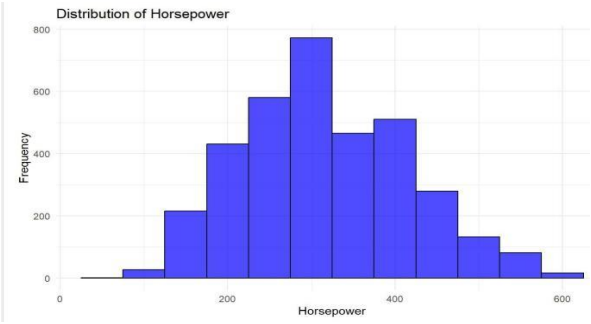

Figure 5. Distribution of Horsepower

```
   brand              model             model_year       milage
Length:3513        Length:3513        Min.   :1992   Min.   :   100
Class :character   Class :character   1st Qu.:2012   1st Qu.: 27000
Mode  :character   Mode  :character   Median :2017   Median : 56842
                                      Mean   :2015   Mean   : 65562
                                      3rd Qu.:2020   3rd Qu.: 95170
                                      Max.   :2024   Max.   :200000
 fuel_type          transmission         ext_col            int_col
Length:3513        Length:3513        Length:3513        Length:3513
Class :character   Class :character   Class :character   Class :character
Mode  :character   Mode  :character   Mode  :character   Mode  :character

  accident          clean_title           price          horsepower
Length:3513        Length:3513        Min.   :  2000   Min.   : 70.0
Class :character   Class :character   1st Qu.: 16995   1st Qu.:245.0
Mode  :character   Mode  :character   Median : 29748   Median :305.0
                                      Mean   : 33192   Mean   :317.7
                                      3rd Qu.: 45000   3rd Qu.:390.0
                                      Max.   : 98000   Max.   :605.0
   cylinders      engine_liter     trans_cat
Min.   :6.000   Min.   :0.650   Length:3513
1st Qu.:6.000   1st Qu.:2.500   Class :character
Median :6.000   Median :3.500   Mode  :character
Mean   :6.502   Mean   :3.626
3rd Qu.:7.220   3rd Qu.:4.600
Max.   :8.943   Max.   :7.400
```

Figure 6. Summary Statistics of Dataset Variables

We're now done with the quick data. Patterns can be seen in the information by looking at how the numerical variables are spread out. The distribution of horsepower looks like a bell, which means that most vehicles are close to the average

horsepower number. There is a right-skewed distribution of prices, with most vehicles priced below $50,000 and some priced higher, which could mean that there are luxury or specialized cars in the market. The distribution of mileage is negatively skewed, meaning that most cars have middling mileage and only a few have high mileage. In terms of associations, price has a strong positive relationship with horsepower (0.60), which means that cars with more powerful engines usually cost more. Also, there is a strong link between cylinders and engine_liter (0.84), which suggests that engine size and cylinder count are related in a straight line. These trends show how prices and results change over time in the dataset.

3. Unsupervised Learning

We used the unsupervised learning techniques to find the patterns and relationships in the used car dataset for our research goal of understanding the structure of the data. By clustering the dataset using various methods, we tried to identify meaningful groupings that could provide insights into price segmentation, customer preferences, and vehicle characteristics, which could later enhance supervised modeling for price prediction. Before we apply the clustering algorithms, we can use the PCA for preprocessing.

3.1. Principal Component Analysis (PCA)

Since we have multiple high dimensional data, we need to use PCA as a tool to reduce the dimensionality of the data to summarize the dataset's variability and identify dominant patterns.

```
Importance of components:
                          PC1    PC2     PC3     PC4    PC5     PC6     PC7     PC8     PC9
Standard deviation     1.6010 1.3280 0.71819 0.65612 0.5788 0.56669 0.51030 0.50317 0.47160
Proportion of Variance 0.3091 0.2127 0.06221 0.05192 0.0404 0.03873 0.03141 0.03053 0.02682
Cumulative Proportion  0.3091 0.5218 0.58405 0.63597 0.6764 0.71510 0.74650 0.77704 0.80386
                          PC10    PC11    PC12    PC13    PC14    PC15    PC16    PC17    PC18
Standard deviation     0.45516 0.44390 0.41610 0.40289 0.35766 0.34086 0.32499 0.29045 0.28089
Proportion of Variance 0.02499 0.02376 0.02088 0.01958 0.01543 0.01401 0.01274 0.01017 0.00952
Cumulative Proportion  0.82885 0.85261 0.87349 0.89307 0.90850 0.92251 0.93525 0.94542 0.95494
```

Figure 7. Summary of Principal Component Analysis (PCA) Variance Contribution

	PC1	PC2	PC3	PC4
model_year	1.847014e-01	-0.6316855058	-0.3293938869	5.345204e-01
milage	-2.275258e-01	0.6134051679	-0.0850578975	5.410379e-01
horsepower	5.781502e-01	-0.0217600525	0.2847210918	1.193871e-01
cylinders	5.657080e-01	0.1971759327	-0.0181835734	-3.405206e-02
engine_liter	4.981042e-01	0.3544278714	-0.2552395279	-2.291420e-02
fuel_E85_Flex_Fuel	2.236007e-03	0.0205604808	-0.0441792301	3.181743e-02
fuel_Gasoline	-2.372151e-02	0.0242453395	0.0859869193	-1.052093e-01
fuel_Electric	1.384481e-02	-0.0488949839	-0.0001313034	4.567911e-02
fuel_Hybrid	-2.090618e-03	-0.0111272024	0.0006775297	7.464634e-03
fuel_Diesel	9.731311e-03	0.0152163660	-0.0423539155	2.024814e-02
clean_title_Yes	-1.417175e-02	0.0864229772	0.0300157906	7.663152e-03
clean_title_No	1.417175e-02	-0.0864229772	-0.0300157906	-7.663152e-03
accident_AtLeast1	-4.244421e-02	0.0984531219	-0.0425196883	2.903254e-01
accident_None	4.244421e-02	-0.0984531219	0.0425196883	-2.903254e-01
trans_Automatic	2.070895e-02	-0.0234069402	-0.1414136752	2.151475e-01
trans_Other	9.004734e-03	-0.0038233251	0.1113787182	-1.944415e-02
trans_Manual	-2.055684e-02	0.0315599978	0.0455397477	-1.945220e-01
trans_CVT	-8.968855e-03	-0.0032921854	-0.0149162170	-3.742260e-04
trans_Multi_Speed	-1.746940e-04	-0.0008175867	-0.0007870139	-7.270965e-04
trans_Single_Speed	-1.329205e-05	-0.0002199604	0.0001984402	-8.001827e-05
brand_groupHigh.End	4.987336e-02	-0.0165643458	0.0921082370	-9.017787e-02

Figure 8. PCA Loadings of Major Features on Principal Components

By analyzing the results of the PCA, we found that retaining the first 18 principal components (PCs) captures 95% of the total variance in the dataset, which balances dimensionality reduction and information retention. The first few components, particularly PC1 and PC2, explain the most substantial variance (30.9% and 21.27%, respectively) and likely reflect key numerical features like mileage, engine related elements, and age of the car, as well as dominant categorical features such as the most frequent fuel or transmission types. This dimensionality reduction simplifies the dataset from its original size to 18 uncorrelated components, preserving almost all meaningful variation while removing redundancy and noise to explain 95% of the variance.

PC1 represents a performance-oriented component, contrasting high-powered, larger-engine, newer cars with low-mileage against older, lower-powered, high-mileage cars. This component aligns well with pricing factors, where higher PC1 values likely correspond to premium or performance cars. PC2 represents an age and usage component, distinguishing older, high-mileage cars from newer, lower-mileage cars. It captures the wear-and-tear aspect of vehicles, often associated with depreciation. These two are the most representative PCs from the result, together explaining about half of the variance.

3.2. K-Means Clustering

Our first algorithm is the K-means clustering. We will use the k-means clustering group cars into clusters based on their PCA features. The first step we need to do is to use the elbow plot to choose the optimal K.

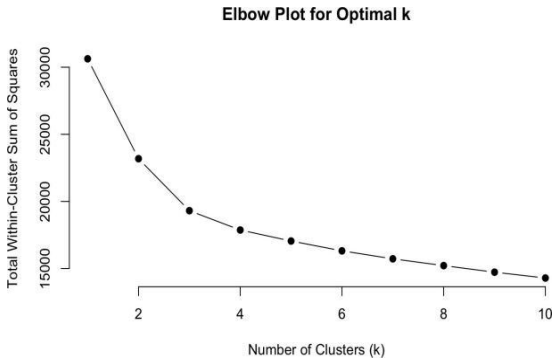

Figure 9. Elbow Plot for Optimal Number of Clusters (k)

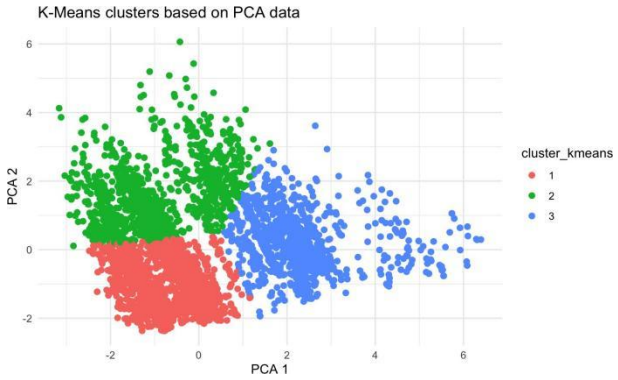

Figure 10. K-Means Clustering Results Based on PCA-Reduced Data

The optimal number of clusters, determined using the elbow method, was around 3, suggesting us to separate the whole dataset into 3 clusters. Using the PCA loadings and

clustering results, we can infer cluster-level characteristics:

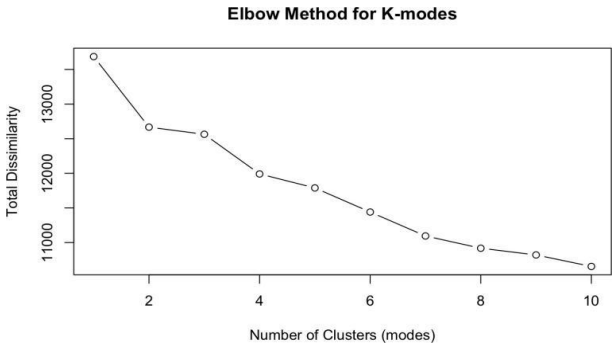

Figure 11. Elbow Method for K-Modes Clustering

Cluster 1 (Red): Likely consists of older, lower-performance cars with higher mileage (negative PC1 and PC2 scores), and they are likely budget-friendly options. Cluster 2 (Green): Likely represents a middle ground with a mix of features. Cars in this cluster might have moderate performance metrics and usage. Cluster 3 (Blue): Likely includes newer, high-performance cars with premium features (positive PC1 scores). Likely higher-priced vehicles, as indicated by correlations with engine size, horsepower, and brand grouping.

3.3. K-Modes Clustering

The K-Modes elbow plot on the left shows that the rate of improvement slows significantly around k = 4, forming an "elbow" point. This suggests that about 4 clusters create a balance between reducing dissimilarity and maintaining interpretability. Selecting 4 clusters would likely capture meaningful patterns in the dataset, such as distinct groupings based on categorical attributes like fuel type, transmission, or other variables.

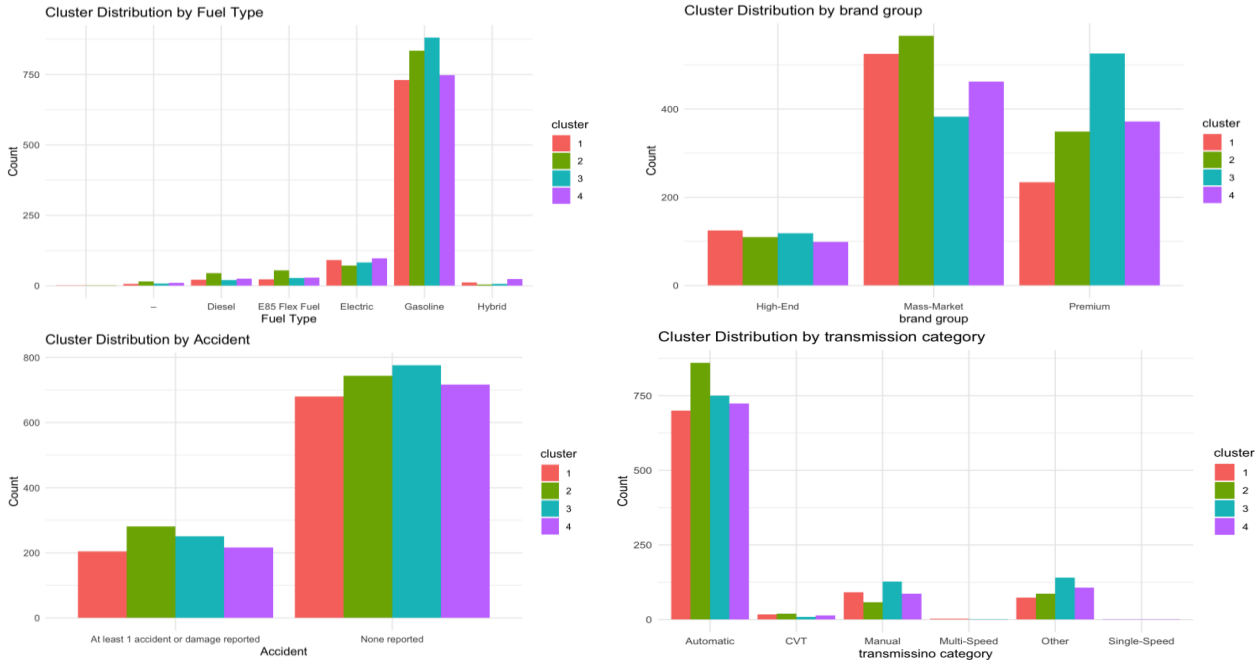

Figure 12. Cluster Distributions by Key Categorical Features in K-Modes Clustering

The K-Modes clustering results reveal distinct groupings of vehicles based on categorical features such as Fuel Type, Brand Group, Accident History, and Transmission Category. Cluster 2, the largest group, is dominated by mass-market vehicles with gasoline engines and automatic transmissions, making it representative of budget-friendly or mainstream options. Cluster 3, which also includes many

gasoline-powered vehicles, features a mix of mass-market and premium cars with no accident history, suggesting it caters to buyers seeking reliable, mid- to high-tier vehicles. Cluster 4 stands out as a niche group, with more diesel, electric, and hybrid vehicles and a notable presence of manual transmissions, reflecting specialty or performance-focused cars. Cluster 1 is smaller and less defined, encompassing a mixed group of vehicles. Cluster 2 may target affordability and broad consumer appeal, Cluster 3 targets quality and reliability, and Cluster 4 highlights niche markets with unique features.

3.4. Hierarchical Clustering

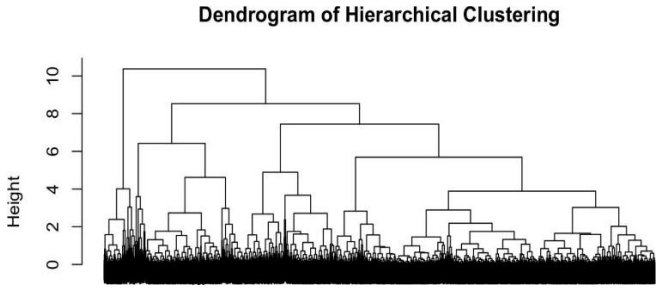

Figure 13. Dendrogram of Hierarchical Clustering

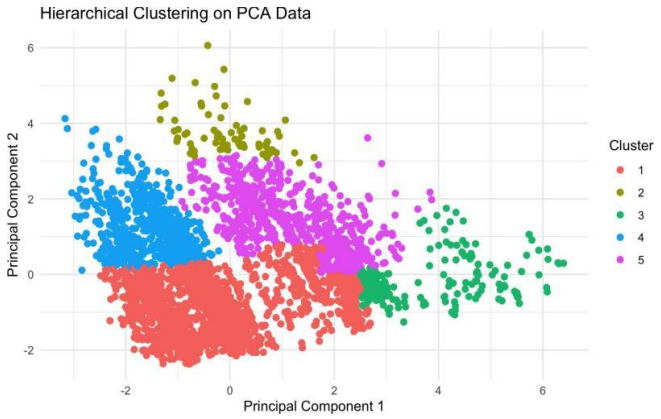

Figure 14. Hierarchical Clustering Visualization on PCA-Reduced Data

The height in the dendrogram represents the dissimilarity at which clusters are joined, and cutting the tree at a height around 5-6 produces approximately 5 clusters, which is related to the scatter plot visualization. The dendrogram shows the hierarchical nature of the clustering process, where smaller groups of similar data points are merged first, followed by increasingly dissimilar clusters as the height increases.

The scatter plot illustrates the clustering results on PCA-reduced data, revealing five distinct clusters. Each cluster corresponds to vehicles with specific characteristics. For example, Cluster 1 (red) at the lower left likely represents budget vehicles, while Cluster 3 (green) on the right includes high-end or premium vehicles. Cluster 2 (yellow) at the top appears to capture niche or specialty vehicles, while Cluster 4 (blue) and Cluster 5 (pink) represent mainstream groups with intermediate features. The clear separation of clusters in the PCA space demonstrates that hierarchical clustering can separate the dataset into meaningful categories, providing information to our further analysis.

4. Prediction Models

Following Unsupervised Learning, we applied the findings to arrange the high-dimensional categorical variables—brand, ext_col, int_col, and model—into more approachable forms. Based on market positioning, we divided the brand variable into three categories: High-End, Premium, and Mass-Market. Using keyword-based grouping to simplify complexity and maintain interpretability, we similarly classified ext_col (external color) and int_col (inside color) into more general tones like Black Shades, White Shades, and Gray Shades. Given its great complexity, the model variable was clustered into four groups using the k-modes algorithm, an unsupervised learning method designed for categorical data, based on numerical characteristics such as cylinders, engine liter, and mileage, as well as categorical characteristics including engine elements, brands, accident/non-accident status, or clean title. We used one-hot encoding to translate these variables into dummy variables, therefore simplifying their inclusion into the supervised learning process once they had been arranged into more general groups. This approach preserved significant information while simplifying the data.

4.1. Dataset

The final dataset consists of six numerical variables, including the response variable price, along with cylinders, model_year, horsepower, and engine_liter. For categorical variables, we have fuel_type with five distinct categories, accident with two categories, and transmission featuring six categories. The brand variable has been grouped into three categories based on market tiers, while ext_col (exterior color) and int_col (interior color) are grouped into eleven and twelve thematic color categories, respectively. Additionally, the model_cluster variable, derived from unsupervised learning, includes four clusters, offering a simplified representation of the vehicle models.

4.1.1. Modeling procedure

For each model, the dataset was split into 80% training and 20% testing sets. Hyperparameter tuning was conducted during training using techniques such as grid search or cross-validation to optimize model parameters. Each model was trained on the training set and evaluated on the test set using metrics such as MAPE, RMSE, MAE, and R^2, ensuring consistency and robustness in comparison.

4.1.2. Evaluation Criteria and Comprehensive Model Assessment

Mean Absolute Percentage Error (MAPE) was our major evaluation metric during model tuning. Particularly appropriate for datasets with a wide range of values, such automobile prices, MAPE was selected since it is unitless and offers an easy % of real value estimate of prediction error. Following the optimal parameter identification, we assessed the final model using other criteria including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R^2. While RMSE gives error values on the same scale as the objective variable, hence allowing improved interpretability, MSE stresses larger errors but may be unduly sensitive to outliers. While R^2 shows the amount of variance explained by the model, therefore suggesting its goodness-of- fit, MAE, being less sensitive to outliers, provides a strong estimate of average error. We guaranteed a thorough evaluation of the prediction accuracy and stability of our models by aggregating these measures.

4.2. KNN Model

4.2.1. Model Description

We initially modified the K-Nearest Neighbors (KNN) regression model parameters and used it to forecast vehicle pricing. Depending on the distance between samples, KNN is an instance-based non-parametric model making predictions. To guarantee that all features could be fairly compared in the distance measure, we conducted one-hot encoding on the categorical variables in the data in practice. Euclidean distance is the default distance metric used by KNN; it applies to both encoded categorical variables and continuous variables.

4.2.2. Parameter Tuning

We tuned the K value (number of neighbors) through grid search, with the search range set to odd numbers from 1 to 20. In each parameter combination, we used 5-fold cross validation to evaluate the performance of the model to ensure the generalization ability of the model parameters. Finally, the best number of neighbors selected by the model was K = 13.

4.2.3. Result

```
Best K value: 13
KNN Model Performance:
MSE: 213087498.94
RMSE: 14597.52
MAE: 10681.32
MAPE: 47.73%
R²: 0.51
```

Figure 15. Performance Metrics of the K-Nearest Neighbors (KNN) Model

From the results, the prediction effect of the KNN model is not ideal, with a MAPE of 47.73% and an R^2 of only 0.51. This shows that KNN has limited processing capabilities for high-dimensional data and cannot capture complex data patterns well.

4.2.4. Cause Analysis

The main reasons why the KNN model performs poorly in this task can be attributed to the following points: Curse of Dimensionality: As the number of features increases, data points become sparse in high-dimensional space, and the distances between points tend to be close, resulting in KNN being unable to distinguish different data points effectively.

Limitations of non-parametric models: KNN does not train parametric models but directly uses samples for prediction, so extracting meaningful features of importance from them is impossible.

Category variable processing: We performed one-hot encoding on categorical variables. Although this is a common preprocessing method for KNN, it may increase the dimension of the data, further exacerbating the dimensionality curse.

4.2.5. Distance measure and feature importance

The KNN model uses Euclidean distance as the metric by default to calculate the similarity between samples. However, Euclidean distance may not be sensitive enough to samples with high-dimensional features or sparse features. Additionally, KNN does not directly train model parameters and therefore cannot quantify feature importance like other parametric models such as linear models or XGBoost.

4.2.6. Conclusion

Overall, the KNN model is not an ideal choice for this task, mainly due to the challenges of high-dimensional data and the limitations of the model itself. Although KNN has a certain theoretical value as an intuitive algorithm, on complex data sets, especially when the feature dimension is high or the data has significant nonlinear characteristics, the prediction performance of KNN is often inferior to other more accurate algorithms. Complex models (such as XGBoost or Elastic Net).

4.3. Elastic-Net Regression

4.3.1. Model Description

The Elastic-Net regression is a regularized linear regression model that combines the penalties of both Ridge and Lasso regression. It effectively handles multicollinearity while selecting significant predictors by shrinking less relevant coefficients to zero. The Elastic-Net's penalty is controlled by the parameters α, which determines the mix of Ridge (α=0) and Lasso (α = 1), and λ, which controls the overall penalty strength.

4.3.2. Tuning Process

To tune the Elastic-Net model, we implemented a two-step approach:

Alpha Selection: We performed a grid search over α values ranging from 0 to 1 with a step size of 0.1 to identify the optimal balance between Ridge and Lasso penalties.

Lambda Selection: Using 10-fold cross-validation, we determined the best λ value by minimizing the mean squared error (MSE). This was achieved using the cv.glmnet function, which internally tunes λ based on the given α.

4.3.3. Evaluation Metrics

Several metrics were used to evaluate the Elastic-Net model's performance: Variance Reduction to quantify the improvement over the baseline variance, MAPE to give an interpretable percentage error measure, R-squared (R^2) to evaluate the proportion of variance explained by the model, RMSE and MSE to measure the magnitude of prediction errors During hyperparameter tuning, MAPE was especially selected since it normalizes mistakes relative to real values and is therefore appropriate for datasets such as automobile prices, which can vary greatly. This guarantees a whole picture of the model by reflecting absolute and relative prediction accuracy.

4.3.4. Model Results

```
Best alpha: 1
Best lambda: 95.5606
RMSE: 11117.86
MSE: 123606702
R²: 0.7084231
Variance Reduced: 2.10523e+11
MAPE: 35.44494 %
MAE: 8161.03
```

Figure 16. Performance Metrics of the Elastic-Net Regression Model

4.3.5. Key Variables and Interpretations

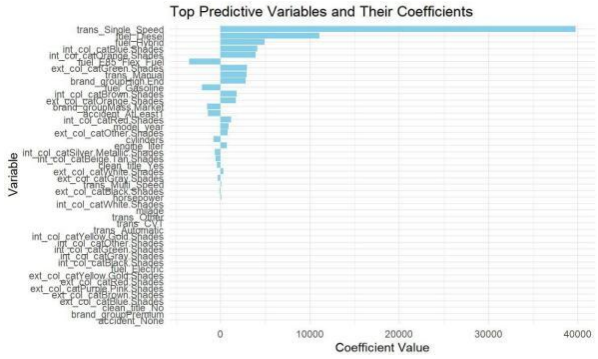

Figure 17. Top Predictive Variables and Their Coefficients in Elastic-Net Model

The coefficient study shows that, presumably in line with their connection with premium cars, Transmission (Single-Speed) and Fuel Type (Diesel) have the most important positive effects on automobile price. Blue Shades and Orange Shades are among the color schemes that also help since they show customer taste for these visual aspects. Furthermore, Cylinders and Engine Liter exhibit modest positive results, which is in line with the hypothesis that bigger engine capacities demand more prices. On the other hand, a history of accidents greatly lowers automobile value, hence underlines the need of vehicle condition in pricing.

4.3.6. Conclusions

With a R^2 of 0.7084, the Elastic-Net model shows great predictive power, therefore explaining 70.8% of the variance in car pricing. An RMSE of 11,117.86 and an MAE of 8,161.03 represent tolerable error magnitudes, therefore supporting the model's accuracy; while, the MAPE of 35.44% shows consistent prediction reliability relative to actual prices. Key predictors—including accident history, fuel type, and gearbox type—offer useful information on the elements driving car costs. The Elastic-Net model is a dependable and understandable solution for this regression problem since thorough adjustment of model parameters guarantees best results.

4.4. Random Forest Prediction Model

4.4.1. Model Description

Random Forest, a popular ensemble learning technique, was implemented to predict the car prices using the ranger

package. This model works by building multiple decision trees during training and averaging their outputs to improve predictive accuracy and control overfitting.

4.4.2. Tuning Process

A grid search was performed over two hyperparameters:

(1) mtry: The number of predictors sampled at each split, which ranged from 2 to the total number of predictors in increments of 2.

(2) nodesize: The minimum size of terminal nodes, evaluated at 5, 10, and 15.

Each combination of these parameters was evaluated using an 80-20 train-test split. The best parameter combination was selected based on the lowest Mean Absolute Percentage Error (MAPE) on the test set.

4.4.3. Results and Feature Importance

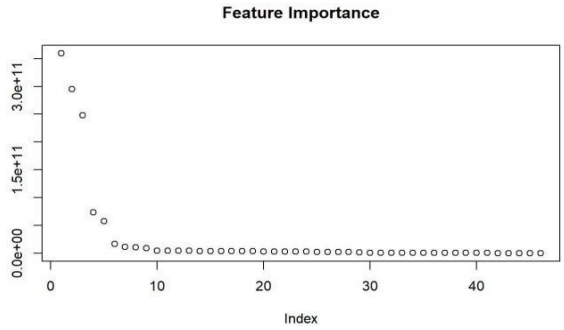

Figure 18. Random Forest Model Performance and Feature Importance Plot

```
Best parameters: mtry = 24, nodesize = 5, Error Rate = 22.60%, R^2 =
0.85
Testing Set Metrics:
MSE: 65378325.91
RMSE: 8085.69
MAE: 5747.31
Error Rate: 22.60%
Variance Reduction: 84.84%
```

Figure 19. Top Predictors Identified by Random Forest Model

4.4.4. The Variable Importance Plot Revealed the Top Predictors of Car Prices

```
```{r}
cat("The important variables:", colnames(X_train)[1:5], "\n")
```

The important variables: model_year milage horsepower cylinders engine_liter
```

From the variable importance, we find that the first five numerical features in the training data are the most important features: model_year, milage, horsepower, cylinders, and engine_liter. These features are the most predictive parameters for determining car prices. To analyze their impact on the price, we used Partial Dependence Plots (PDP), which show how a specific feature affects the predicted price while keeping all other features constant.

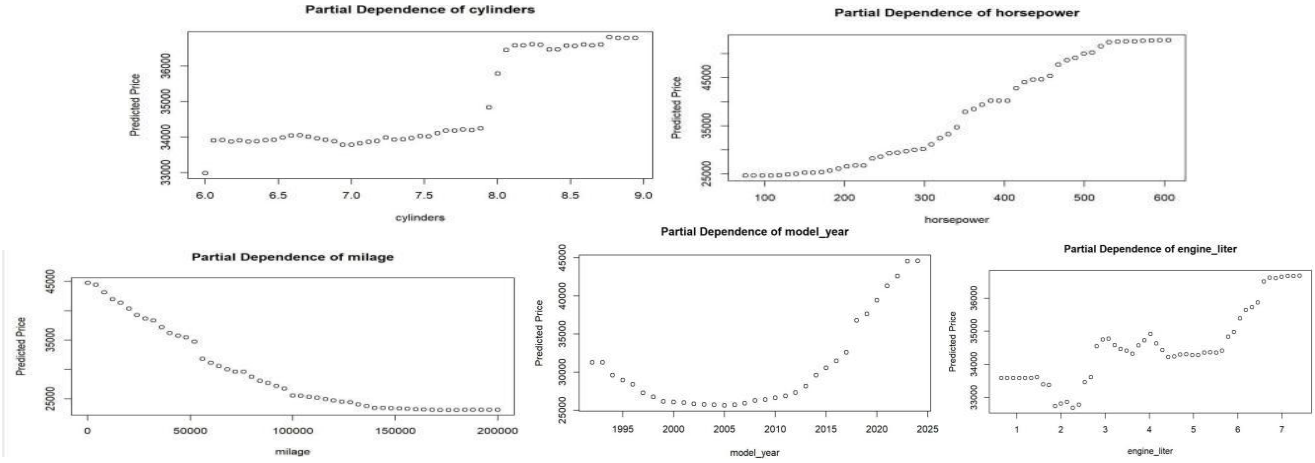

Figure 20. Partial Dependence Plots (PDPs) of Key Predictors in the Random Forest Model

The PDP for model_year shows a positive relationship with the price, indicating that newer cars are valued higher. For milage, the PDP exhibits a negative relationship, reflecting depreciation as the car's mileage increases. Similarly, horsepower and engine_liter both demonstrate positive effects, suggesting that higher engine performance and capacity are associated with higher prices. On the other hand, cylinders show a more nuanced relationship, where fewer cylinders may correlate with lower performance and pricing tiers. These insights confirm the importance of these variables in capturing key patterns in the data and their significant role in price prediction.

4.4.5. Model Insights

Random Forest showed robust performance with an R^2 value of 0.85, explaining 85% of the variance in the car prices. The error metrics (MAPE: 22.60%) indicate that the model provides accurate predictions for most cases but may struggle with extreme outliers.

The hyperparameter tuning ensured that the model achieved a balance between bias and variance, making it well-suited for the given dataset. Features like cylinders and engine size emerged as the most predictive, reaffirming their importance in car pricing.

4.4.6. Discussion on Specific Evaluation Metrics

The performance of the random forest model is evaluated using several metrics: RMSE and MSE are used to measure the size of the prediction error, R-squared (R^2) is used to evaluate the proportion of variance explained by the model, variance reduction is used to quantify the improvement relative to the baseline variance, MAPE is used to provide an explainable percentage error measure, and MAE is used to capture the average magnitude of the absolute errors. We use MSPE as a metric during the parameter tuning process because MSPE enhances the evaluation by normalizing the

errors relative to the actual values, making it particularly suitable for a wide range of datasets such as car prices. This combination of absolute and relative metrics provides a balanced and comprehensive evaluation of the model's predictive accuracy and robustness.

4.4.7. Conclusion

The Random Forest model achieved strong performance with an R^2 of 0.85, explaining 85% of the variance in car prices. Key metrics include an RMSE of 8085.69, MSE of 65378325.91, MAE of 5747.31, and a MAPE of 22.60%, indicating accurate predictions with some sensitivity to outliers. Hyperparameter tuning optimized the balance between bias and variance, with features like milage, cylinders and engine size emerging as the most predictive, demonstrating the model's effectiveness and interpretability.

4.5. XGBoosting Model

4.5.1. Model Description and Tuning Process

We employed a Tree Boosting model, a powerful gradient boosting algorithm, implemented using the caret and gbm packages, to predict car prices based on a mix of numerical and categorical variables such as mileage, engine specifications, model year, and brand group. Features were preprocessed with appropriate scaling and one-hot encoding to ensure compatibility with the boosting framework.

Hyperparameter tuning was conducted using Mean Absolute Percentage Error (MAPE) as the primary evaluation metric due to its interpretability, especially for datasets with widely varying price ranges. A grid search was performed over the following hyperparameters:

(1) n.trees: Number of boosting iterations (100, 200, 300).

(2) interaction.depth: Maximum depth of each tree (1, 3, 5).

(3) shrinkage: Learning rate (0.01, 0.1, 0.2).

(4) n.minobsinnode: Minimum number of observations in a terminal node (10, 20).

Each combination of hyperparameters was evaluated on a validation set (20% split), with MAPE chosen as the primary criterion because it normalizes errors relative to the actual values. This makes it particularly suitable for datasets like car prices that span a wide range. Additional metrics such as Root Mean Squared Error (RMSE) and R^2 were also recorded to assess the overall performance of the model. The Tree Boosting model effectively captured complex feature relationships, resulting in strong predictive performance.

4.5.2. Results

```
Best parameters:
$max_depth
[1] 7

$eta
[1] 0.1

$subsample
[1] 0.8

$colsample_bytree
[1] 0.8

$objective
[1] "reg:squarederror"

$eval_metric
[1] "rmse"

$nrounds
[1] 200

MAPE: 21.07%
MSE: 56480048.69
RMSE: 7515.32
MAE: 5416.08
Variance Reduction (R²): 0.87
```

Figure 21. Performance Metrics and Hyperparameters of the XGBoost Model

The XGBoost model, with optimal parameters (max_depth = 7, eta = 0.1, subsample = 0.8, colsample_bytree = 0.8, and nrounds = 200), achieved strong performance on the test set. Key metrics include a MAPE of 21.07%, MSE of 56,480,048.69, RMSE of 7,515.32, MAE of 5,416.08, and R^2 of 0.87, explaining 87% of the variance in car prices. These results highlight the model's robust predictive accuracy and ability to capture complex feature relationships effectively.

4.5.3. Feature Importance and Insights

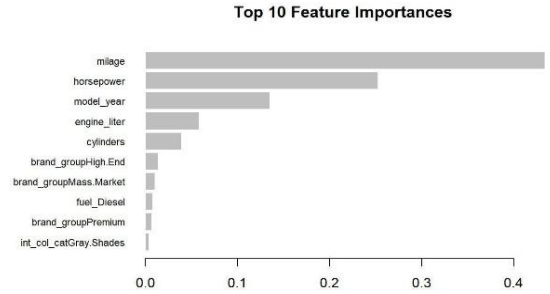

Figure 22. Top 10 Feature Importances in the XGBoost Model

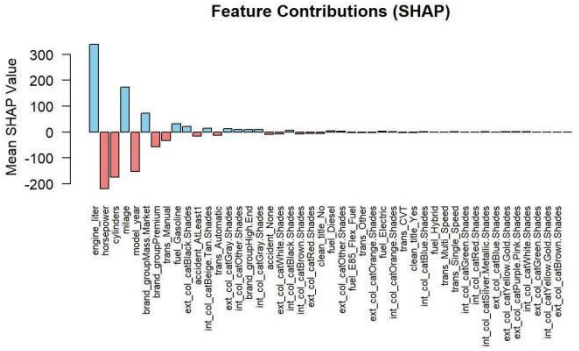

Figure 23. SHAP Summary Plot of Feature Contributions in the XGBoost Model

The following were the top predictive factors found using the XGBoost feature significance function: Mileage turned up as the most important factor since it clearly lowers vehicle prices. Reflecting consumer tastes for more powerful engines, horsepower showed a positive association with price. Model year also was important since newer cars, which reflect declining patterns in depreciation, attract more money. Another key consideration was engine liter capacity since higher engine capacities are linked to premium price. Prices were lowered by cylinders, which represent engine capacity. Among categorical variables, the "High-End" brand group highly suggested premium pricing; "Mass-Market" brands were linked to mid-tier pricing patterns.

Because of its efficiency and longevity, fuel type—especially diesel—was favorably linked with higher pricing.

4.5.4. Discussion of Evaluation Metrics

The evaluation measures R^2, MAPE, and RMSE provide a whole picture of model performance. Selected as the main hyperparameter tuning criterion, MAPE was especially appropriate for datasets with a wide range of prices since it measured relative errors as percentages, therefore providing an easy interpretability. This emphasis on relative performance guaranteed that forecasts were assessed fairly on several price ranges.

Emphasizing general prediction accuracy, RMSE complimented MAPE by recording the standard deviation of residuals, while R^2 measured the fraction of variance explained by the model, hence providing a simple indication of goodness-of-fit. These indicators taken together gave a fair and efficient structure for assessing and improving the model.

4.5.5. Conclusion

The XGBoost model demonstrated excellent performance, with its ability to capture complex interactions between features and maintain high predictive accuracy. The inclusion

of features like mileage, horsepower, and brand group ensured both interpretability and relevance to car pricing dynamics. The hyperparameter tuning process further enhanced the model's generalization, making XGBoost one of the most robust models in this analysis.

4.6. Support Vector Machine

4.6.1. Model description

The Support Vector Machine (SVM) is a flexible and powerful machine learning model that can be used for regression tasks. In this project, I employed the radial basis function (RBF) kernel to capture potential

non-linear relationships between the predictors and the target variable (price). The performance of the model was tuned by optimizing two key hyperparameters: cost (C) and gamma (γ). These parameters help balance the model's complexity and generalizability.

4.6.2. Hyperparameter Tuning and Evaluation Criteria

The Support Vector Machine (SVM) is a flexible and powerful machine learning model used for regression tasks. In this project, we utilized the radial basis function (RBF) kernel to capture potential non-linear relationships between predictors and the target variable (price). To optimize the model's performance, a grid search was conducted over the hyperparameter ranges of cost (C) from 0.1 to 1 (incremented by 0.1) and gamma (γ) from 0.01 to 0.1 (incremented by 0.01). The model was trained on 80% of the dataset and tested on the remaining 20%, and its performance was evaluated using metrics such as Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R^2 (Coefficient of Determination), and Variance Reduction. MAPE was selected as the primary criterion for model selection due to its interpretability and sensitivity to large errors since some prices may be very large which can make big value of MSE, RMSE or MPE, while the other metrics provided a comprehensive understanding of the model's predictive accuracy and ability to capture variability in the target values.

4.6.3. Results And Variable Importance

```
Minimum MAPE: 26.19%
Best parameter combination: cost_1_gamma_0.01
Corresponding R²: 0.76
Variance reduction: 76.38%
Best MSE: 102450381.1854
Best MAE: 6852.8544
Best RMSE: 10121.7776
```

Figure 24. SVM Model Performance

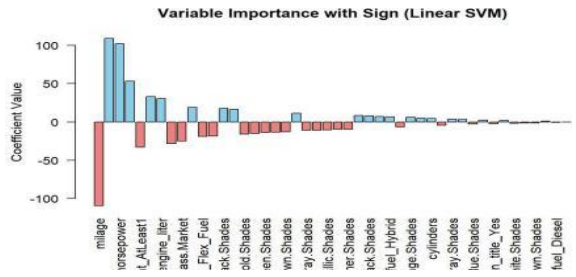

Figure 25. Variable Importance with Coefficient Signs (SVM)

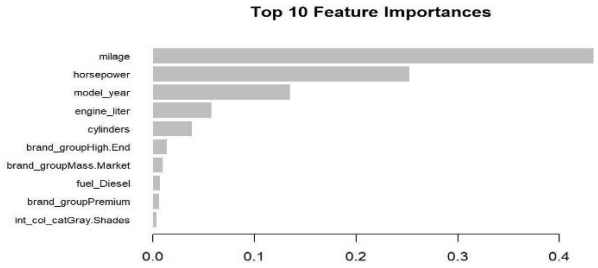

Figure 26. Top 10 Feature Importances in SVM Model

The best SVM model, with Cost = 1 and Gamma = 0.01, achieved a MAPE of 26.19%, MSE of 10,245,038.19, MAE of 6,852.85, RMSE of 10,121.78, and R^2 of 0.76, explaining 76% of the variance in car prices with low prediction error.

Using both SHAP (SHapley Additive exPlanations) values and XGBoost feature importance scores, I identified the most predictive variables and interpreted their impacts on vehicle prices. Mileage was the most important predictor, with SHAP values confirming its negative impact on price due to depreciation from higher usage. Horsepower and engine liter capacity positively influenced prices, reflecting consumer preferences for powerful and premium vehicles. Model year was another key factor, as newer cars generally command higher prices due to lower depreciation. Cylinders showed a slight negative impact, suggesting diminishing returns for larger engines beyond a certain point. Among categorical variables, "High-End" brand groups were associated with premium pricing, while "Mass-Market" brands aligned with mid-tier pricing. Additionally, fuel type, particularly diesel, had a positive effect due to its efficiency and durability. This combined analysis provided a clear understanding of both the magnitude and direction of each variable's influence on vehicle pricing.

4.6.4. Conclusion

The SVM model with the RBF kernel achieved strong predictive performance with a minimum MAPE of 26.19% and an R^2 of 0.76. The most important features were mileage, horsepower, and engine size, which align with industry expectations about the factors influencing car prices. While computational cost and hyperparameter tuning remain challenges, the model's flexibility and predictive power make it a valuable tool for regression tasks in this domain.

4.7. Overall Conclusion for the Prediction Models

Judging from the results of the five models, the MAPE of the Elastic Net model is 35.44% and the R^2 is 0.708, indicating that linear information has limited ability to explain the overall data; the MAPE of the KNN model is as high as 47.73% and the R^2 is only 0.51. It is further shown that its performance is poor in high-dimensional space. This also implies the presence of significant nonlinear trends in the data.

Among nonlinear models, Random Forest, SVM, and XGBoost exhibit stronger fitting capabilities. The MAPE of the random forest model is 22.60% and R^2 is 0.85, showing strong nonlinear feature modeling ability; the MAPE of the SVM model is 26.19% and R^2 is 0.76, which can also capture non-linear features in the data to a certain extent. linear relationship. However, the XGBoost model performed the best, achieving the lowest MAPE (21.07%) and the highest R^2 (0.87) on the test set, indicating that it can more effectively capture nonlinear patterns in the data. In summary, the XGBoost model has the strongest nonlinear modeling capabilities, can effectively capture complex feature

relationships, and further optimizes model performance through hyperparameter tuning. Therefore, XGBoost is the optimal prediction model in this task.

4.7.1. Feature Importance and Variable Impact Analysis

Judging from the results of feature importance and variable influence, although the characteristic coefficients of the Elastic Net model provide some linear information, due to its low R^2 and large error, we speculate that the linear information in the data is limited, and its variables Importance has low credibility. Since the KNN model does not train a parameterized model, it cannot extract clear feature importance. In nonlinear models, Random Forest and XGBoost provide more reliable feature importance analysis. As can be seen from the Feature Importance and SHAP value plots of XGBoost, features such as horsepower, mileage, and model_year have the greatest impact on predictions. Among them, horsepower and model_year have a positive impact on price, indicating that consumers are more inclined to pay a premium for vehicles with more power and newer years; while mileage is negatively related to price, reflecting that the more miles the vehicle has traveled, the higher its residual value. Low. Brand classification features (such as brand_groupPremium and brand_groupHigh. End) also have a significant positive effect on price, further indicating that the market positioning of high-end brands enhances the nonlinear characteristics of vehicle pricing. The importance of these nonlinear characteristics illustrates that capturing nonlinear patterns is critical for accurate price predictions.

From the comprehensive performance of the five models, XGBoost shows the best nonlinear modeling ability with its lowest MAPE and highest R^2, and its feature importance analysis also provides a more reliable variable explanation. This shows that nonlinear features are the main driving force of this data analysis. The XGBoost model can clearly reveal the complex relationships in the data and achieve the best prediction results.

5. Open-Ended Question

5.1. Solution Approach

To estimate the original price of cars in our dataset as if they were brand new, we used predictive modeling combined with data adjustment techniques:

(1) Predictive Models: We employed Random Forest and XGBoost, both of which demonstrated strong performance in earlier evaluations, to generate predictions for adjusted car attributes.

(2) Data Adjustments: Three cars were randomly selected, and their data was modified to simulate new conditions:

model_year was updated to 2025 to reflect the latest model year.

mileage was set to 100, representing a brand-new vehicle.

All other attributes (e.g., brand, horsepower, fuel type) were kept unchanged to retain the car's unique features.

(3) Price Predictions: Using the adjusted data, price predictions were generated with both models, and these predictions were compared with the actual release prices retrieved from car manufacturers' websites.

5.2. Challenges and Limitations

Several challenges emerged during this process:

(1) Extrapolation Risks: The models were trained on a dataset of used cars, where price patterns reflect depreciation. Predicting prices for new cars required extrapolation, which is less reliable due to differences in pricing dynamics.

(2) Data Gaps: The dataset lacked critical information influencing new car prices, such as market-specific promotional discounts, additional optional features, and regional pricing strategies.

(3) Model Bias: The representation of certain brands or car types in the dataset may have been imbalanced, potentially affecting model accuracy.

5.3. Results and Comparison

We selected three cars for analysis, adjusted their attributes to simulate new vehicles, and compared the model predictions with official release prices:

(1) Ford LARIAT:

Predicted Prices:

Random Forest: $69,185.31

XGBoost: $61,607.81

Official Price: $63,260

Analysis: Random Forest slightly overestimated, while XGBoost underestimated. These differences could stem from unobserved optional features or dealership pricing incentives.

(2) Cadillac CT5:

Predicted Prices:

Random Forest: $50,287.25

XGBoost: $50,939.36

Official Price: $48,990

Analysis: Both models were highly accurate, with minor overestimations likely caused by factors such as unaccounted regional pricing or promotional variations.

(3) Kia Niro Plug-in Hybrid EX:

Predicted Prices:

Random Forest: $32,680.46

XGBoost: $37,640.06

Official Price: $34,490

Analysis: Random Forest slightly underestimated, while XGBoost slightly overestimated, potentially reflecting challenges in accounting for hybrid-specific features or market positioning.

5.4. Conclusion

The predictions closely aligned with the official prices, demonstrating the efficacy of our models in estimating new car prices. However, discrepancies arose due to unobserved factors, such as optional packages and regional pricing strategies. Future improvements could include incorporating additional data on new car features, regional price variations, and market incentives to enhance prediction accuracy.

References

[1] Hankar, M., Birjali, M., & Beni-Hssane, A. (2022). Used Car Price Prediction using Machine Learning: A Case Study. 2022 11th International Symposium on Signal, Image, Video and Communications (ISIVC), 1–4. https://objects.scraper.bibcitation.com/user-pdfs/2024-12-12/849adcb3-3f87-41e3-b337-290735fd f4f1.pdf

[2] Gegic, E., Isakovic, B., Keco, D., Masetic, Z., & Kevric, J. (2019). Car Price Prediction using Machine Learning Techniques. TEM Journal, 8(1), 113–118. https://doi.org/10.18421/tem81-16

[3] Stekhoven, D. J., & Bühlmann, P. (2011). MissForest—nonparametric missing value imputation for mixed-type data. Bioinformatics, 28(1), 112–118. https://doi.org/10.1093/bioinformatics/btr597